

CLAIM AMENDMENTS

Claim Amendment Summary

Claims pending

- Before this Amendment: Claims 1-2 and 5-19 and 21-27.
- After this Amendment: Claims 1-2 and 5-14

Non-Elected, Canceled, or Withdrawn claims: 15-19 and 21-27

Amended claims: None

New claims: None

Claims:

1. (Previously Presented) A computer-implemented method for generating a chaos-based pseudo-random sequence (X_n) comprising the steps of:

defining a chaotic map for generating a pseudo-random sequence of integer numbers (x_n) comprised in a certain interval $[0, q]$;

defining a function ($H(x)$) on a first interval $x \in [0, q]$ whose inverse has a plurality of branches;

choosing a seed (x_0) of said pseudo-random sequence of integer numbers (x_n) comprised in said interval $[0, q]$;

generating numbers of said pseudo-random sequence (x_n);

calculating numbers of a chaos-based pseudo-random sequence (Xn) by applying said function (H(x)) to corresponding integer numbers of said pseudo-random sequence (xn);

storing said chaos-based pseudo-random sequence (Xn) in a circuit;
and

generating encrypted data, using a computer-implemented data generator, on a computer-readable medium by utilizing said chaos-based pseudo-random sequence (Xn) in an encryption application executing on a computer;

wherein said chaotic map is a linear congruential generator defined by:

choosing a first integer number (m);

choosing a second odd integer number (p) greater than the power of 2 raised to said first integer number (2m);

choosing a third integer number (M) much greater than said first integer number (m);

said chaotic map being defined by the following equation:

$$x_{n+1} = \left(\frac{p}{2^m} \cdot x_n \right) \bmod 2^M.$$

2. (Original) The method of claim 1, wherein the inverse of said function (H(x)) has a number of branches equal to the largest bound (q) of said interval ([0, q]).

3. (Canceled)

4. (Canceled)

5. (Original) The method of claim 1, wherein defining said function (H(x)) comprises defining (H(x)) such that it may assume only two values ().

6. (Original) The method of claim 5, comprising the steps of:
representing in binary form said integer numbers (x_n) of said pseudo-random sequence;
defining a second integer number k;
defining said function (H(x)) as the binary sum of the k least significant bits of the binary representation of its argument (x).

7. (Original) The method of claim 5, wherein said chaotic map is a truncated linear congruential generator.

8. (Original) The method of claim 7, wherein said truncated linear congruential generator is defined by:

choosing a first integer number (m);
choosing a second odd integer number (p) greater than the power of 2 raised to said first integer number (2^M);
choosing a third integer number (M) much greater than said first integer number (m);

said chaotic map being defined by the following equation:

$$x_{n+1} = \text{trunc}_k \left(\left(\frac{p}{2^m} \cdot x_n \right) \bmod 2^M \right).$$

9. (Original) The method of claim 7, wherein said linear congruential generator is defined by:

choosing a first integer number (m);

choosing a second odd integer number (p) greater than the power of 2 raised to said first integer number (2m);

choosing a third integer number (M) much greater than said first integer number (m);

said chaotic map being defined by the following equations:

$$\begin{cases} y_n = x_n \oplus X_n \\ x_{n+1} = \text{trunc}_k \left(\left(\frac{p}{2^m} \cdot y_n \right) \bmod 2^M \right) \end{cases}$$

10. (Previously Presented) The method according to claim 1 wherein said third integer number (M) is greater than or equal to 64.

11. (Previously Presented) The method of claim 6, comprising the steps of:

providing circuit means for storing bit strings representing integer numbers (x_n) of said pseudo-random sequence;

providing a shift register (R1) coupled to said circuit means;

storing a seed (x₀) in said circuit means;

carrying out cyclically the following operations:

copying in said shift register (R1) a bit string stored in the circuit means representing a current number (x_n) of said pseudo-random sequence,
providing k shift commands to said shift register,
generating a bit (X_n) of said chaos-based pseudo-random bit sequence by summing modulo 2 the k bits output by said shift register,
generating a bit string representing a successive number (x_{n+1}) of said pseudo-random sequence by summing up the bit string currently stored in said shift register (R1) and the bit string representing said current number (x_n),
storing in the circuit means the bit string representing said successive number (x_{n+1}).

12. (Previously Presented) The method of claim 6, comprising the steps of:

providing circuit means for storing bit strings representing integer numbers (x_n) of said pseudo-random sequence;

providing a register $[(R1)]$ coupled to said circuit means;

storing a seed (x_0) in said circuit means;

carrying out cyclically the following operations:

copying in said register a bit string stored in the circuit means representing a current number (x_n) of said pseudo-random sequence,

generating a bit (X_n) of said chaos-based pseudo-random bit sequence by summing modulo 2 the k least significant bits of the bit string stored in said register,

generating a bit string representing a successive number (x_{n+1}) of said pseudo-random sequence by summing up the bit string representing said current number (x_n) and the bit string obtained eliminating the k least significant bits of the bit string stored in said register,

storing in the circuit means the bit string representing said successive number (x_{n+1}).

13. (Previously Presented) A generator of chaos-based pseudo-random bit sequences operable in an encryption application, comprising:

circuit means for storing bit strings representing integer numbers (x_n) of said pseudo-random sequence;

a register coupled to said circuit means;

an adder modulo 2 summing the k least significant bits of the of the bit string stored in said register, generating a bit (X_n) of said chaos-based pseudo-random bit sequence;

a second adder summing up the bit string representing said current number (x_n) and the bit string obtained eliminating the k least significant bits of the bit string stored in said register; and

a data generator for generating encrypted data on a computer-readable medium based on said pseudo-random sequence.

14. (Previously Presented) A generator of chaos-based pseudo-random bit sequences operable in an encryption application, comprising:

circuit means for storing bit strings representing integer numbers
(x_n) of said pseudo-random sequence;

a shift register coupled to said circuit means;

a command circuit generating shift commands for said shift register;

second circuit means for storing the bits output by said shift register;

an adder modulo 2 summing the bits stored in said second circuit
means, generating a bit (X_n) of said chaos-based pseudo-random bit sequence;

a second adder summing up the bit strings currently stored in said
shift register and in said first circuit means, generating a bit string representing a
successive number (x_{n+1}) of said pseudo-random sequence; and

a data generator for generating encrypted data on a computer-readable
medium based on said pseudo-random sequence.

15. – 27. (Canceled)